https://doi.org/10.2352/ISSN.2470-1173.2019.8.IMAWM-415

Non-native Contents Detection and Localization for Online Fashion Images*

Litao Hu, Karthick Shankar, Zhi Li, Zhenxun Yuan, Jan Allebach; Purdue University; West Lafayette, IN
Gautam Glowala, Sathya Sundaram, Perry Lee; Poshmark Inc.; Redwood City, CA

Abstract

Non-native content detection is about detecting regions of contents in an image that do not belong to the original or natural contents of the image. In the online fashion market, sellers often add non-native contents to their product images in order to emphasize the features of their products and get more views. However, from the buyer's point of view, these excessive contents are often redundant and may interfere with the evaluation of the major contents or products in the image. In this paper, we propose two methods for detecting non-native content in online fashion images. The first one utilizes the special properties of image mosaicing and de-mosaicing where there are local correlations between pixels of an image. The second method is based on the periodic properties of interpolations which is a common process involved in the creation of forged images. Performance of the two methods are compared by testing on a dataset consisting of real images from an online fashion marketplace. The experimental results demonstrate the effectiveness of both methods.

Introduction

Nowadays image forgery has become a very common phenomenon in this web and mobile world; and it is getting much easier for people to create forged images with their mobile devices. Therefore, being able to detect and weed out the forged images is very important in a lot of cases.

Based on the techniques used in the process of image forger, types of image forgery can be roughly divided in three classes, namely copy-move forgery, where a subset of an image is copied and pasted in another location in the same image; image splicing, where a subset of an image is copied and pasted to another image; and lastly image re-sampling, where a region of an image is enhanced by geometric transformations like rotation, skewing, and scaling. In this paper, we will mainly be concerned with image splicing and re-sampling.

We propose two methods for non-native contents detection in this paper. In the first method, primarily intended for non-native text detection, a color image is used to identify maximally stable extremal regions. We then filter these regions to get possible text regions. After that, we mosaic and demosaic the image using the Bayer color filter array to generate demosaicing artifacts. These artifacts are caused by local correlations between pixels in a color image due to the demosaicing algorithm of the camera used to take the original picture. Finally, we perform morphological operations to generate an error image which can be used to localize the non-native content. An illustration of this process is shown in Figure 1.

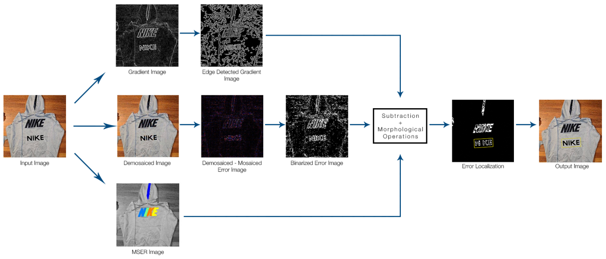
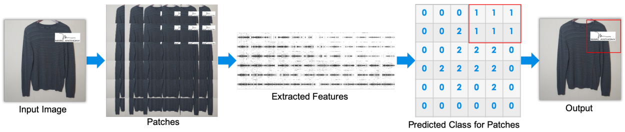


Figure 1: Pipeline of the First Method

In the second method, an gray-scale input image be used to generate multiple smaller patches, with a chosen step size. Then a feature vector will be extracted for each patch. After that, we use a trained Siamese classifier to classify each pair of patches and therefore classify patches into different groups. Finally, the existence of non-native contents will be determined based on the number of groups and located as the group with the least number of patches. An illustration of the process is shown in Figure 2.



0	0	0	1	1	1
0	0	2	1	1	1
0	0	2	2	2	0
0	2	2	2	2	0
0	0	2	0	2	0
0	0	0	0	0	0

Figure 2: Pipeline of the Second Method

In the following sections, we will introduce some related work on detection of image forgery and non-native contents. Then, we will describe in details the two non-native contents detection and localization algorithms, along with their key building blocks. Finally, experimental results comparing the two methods will be presented, followed by a brief conclusion.

Related Work

There are numerous research papers and proposed methods for image forgery detection in the field of image forensics. Conventional methods such as [1, 2, 3, 4, 5] are mostly based on detecting traces of resampling processes like linear interpolation and cubic interpolation. Reference [5] introduced a method to calculate features for image patches using periodic properties of interpolation, which is adopted and improved in the second method proposed in this paper to calculate feature vectors for classification.

Many recent methods, such as [6, 7, 8], take advantage of deep learning models like convolutional neural networks (CNN) and long-short term memory (LSTM). In [8], two methods utilizing different deep learning models were introduced. In the first

*Research partially supported by Poshmark, Inc., Redwood City, CA 94065

method, overlapping patches are extracted from the input image and then classified by six distinct fully connected neural networks, each corresponding to a distinct resampling process. After that, six pixel-level classification maps will be generated by the six classifiers, which will then be considered together to form a final mask. In the second method of [8], an LSTM framework is used to perform patch classification, from which a final mask will be generated for an input image.

Proposed Methods

In this paper, we introduce two methods for detecting and localizing image non-native contents from different perspectives.

Method 1: Detection and Localization Using Image Mosaicing Artifacts

In the context of online fashion, non-native text is abundant in terms of fake branding and buzz-words like ”sale” or ”discounts”. Thus, by focusing on text-detection, we restrict the domain of non-native regions and simplify the detection process.

Text Region Detection

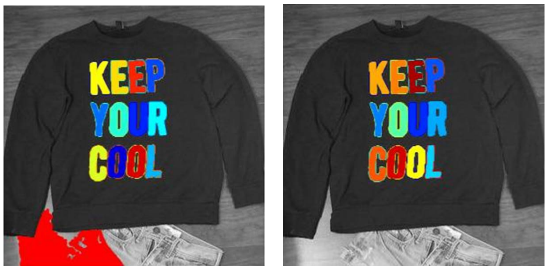
As outlined in [9], extremal regions of an image possess highly desirable properties, namely the set is closed under continuous (and thus projective) transformation of image coordinates, as well as the monotonic transformation of image intensities. Thus, MSERs or Maximally Stable Extremal Regions are the regions that are defined solely by an extremal property of the intensity function in the region and on its outer boundary. It is a stable connected component of some gray-level region of an image, where stability is defined as having virtually unchanged gray-levels over a range of thresholds of a binary image [9].

Given a source image, we generate a sequence of thresholded result images I_t where each image t corresponds to an increasing threshold t. First, we get a white image, then we get ’black’ spots corresponding to local intensity minima which grow larger. These ’black’ spots merge until the whole image is black. The set of all connected components in the sequence is the set of all extremal regions [9]. Text regions are generally stable due to even coloring and consistent fonts [10]. The MSERs of a sample image from an online fashion store are shown in Figure 3a.

However, non-text regions can also have high stability as shown by the red region on the bottom left of Figure 3a. We use a rule-based approach to remove these non-text regions [11]. Geometric properties of text are used to filter out non-text regions using simple thresholds [12]. The properties are:

1. Aspect Ratio: Dimensions of the smallest rectangle containing the region, returned as a $1 \times Q$ vector, where Q is the number of image dimensions.
2. Eccentricity: An ellipse is drawn around the region. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length.
3. Euler Number: Number of objects in the region minus the number of holes in those objects.
4. Extent: Ratio of pixels in the region to pixels in the total bounding box.
5. Solidity: Proportion of the pixels in the convex hull that are also in the region.
6. Stroke Width: A measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations [13].

The filtered image is shown in Figure 3b.

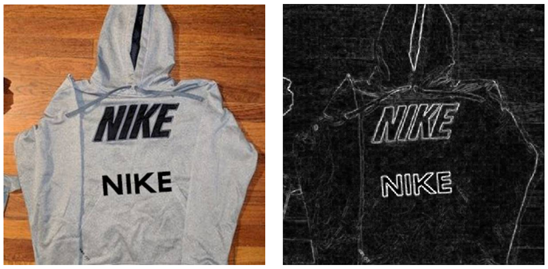


(a) MSERs (colored regions) of sample image (b) MSERs after filtering with geometric properties

Figure 3: MSERs

Image Preprocessing

Once we have the text regions, we preprocess the image. In any image, the text portion has distinct gray-values with respect to the non-text region or background. This difference in gray-values is captured in a gradient image, as outlined in [14]. Morphological gradient operations of erosion and dilation are applied on the input image. The input image and its corresponding gradient image is shown in Figure 4 After obtaining the gradient image, we apply an edge detector and binarize the image [14]. The method in [14] uses the Sobel edge detector. But in our experience this was not able to preserve a lot of the detail in the image, as shown in Figure 12. This is because the Sobel edge detector was used only with two directions vertical and horizontal [15]. All other directions of edges were not considered, which induced the loss in detail. Hence, we use the Canny edge detector which can detect a wide range of edges in images [16].



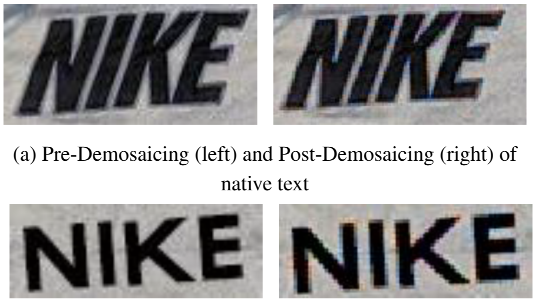
(a) Input Image with fake branding (bottom Nike word) (b) Gradient Image after morphological operations

Figure 4: Input and Gradient Image

Mosaicing and Demosaicing

Images captured by a camera have certain intrinsic properties which are caused by the demosaicing algorithm the camera uses. There are local correlations among the pixels of the original image which are caused by the process of demosaicing [17]. A camera generally only captures one of the 3 components of color through its sensor and estimates the values of the other components of that pixel using the component values of its neighboring pixels [18].

Hence, a demosaicing algorithm reconstructs a full color image from the incomplete color samples output from an image sensor with the help of a Color Filter Array (or CFA) [19]. In the case of online fashion shopping, we do not have an original image for comparison. Thus, we recreate this process by applying the inverse of demosaicing to a candidate image. Mosaicing an image is a process in which we discard two of the three components of a given pixel. The choice of component to be kept is decided by the color filter array that is used. The Bayer filter mosaic is a very common CFA, so that is the filter that we use. We then demosaic the image as outlined in [20]. On applying both these processes to an image, we see that the edges of artificial texts have some discoloring artifacts caused by the local demosaicing correlations. The discoloring is less apparent in regions with natural text. Figure 5 shows two images before and after the demosaicing process. Note that the lower Nike is artificially inserted while the upper Nike is a natural element of the image.

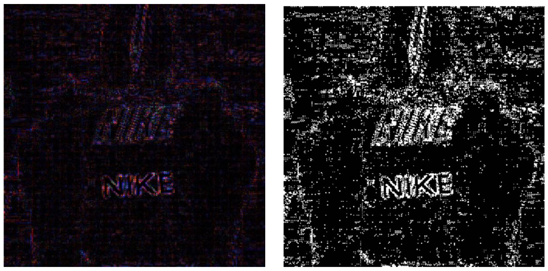


(a) Pre-Demosaicing (left) and Post-Demosaicing (right) of native text

(b) Pre-Demosaicing (left) and Post-Demosaicing (right) of non-native text

Figure 5: Discolorization Comparison

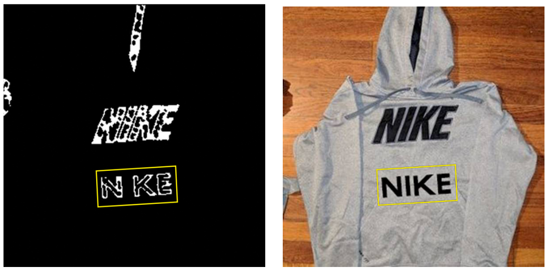
We generate the error image, which is the difference between the demosaiced image and the input mosaiced image, and binarize it as shown in Figure 6. We then superimpose the MSERs onto this binarized error image and subtract the edge detected gradient image (Figure 4 to get the error on the edges since that is where the discolorization exists. On this image, we calculate the amount of discoloring, or ratio of black pixels to white pixels in any given region and use that to determine if a region is artificial or not. If the ratio is lower than a given threshold determined by the edge thickness, it is considered as artificial as shown in Figure 7a. This method works best for classifying an entire image as having non-native content or not, rather than localizing regions containing non-native content since the MSER detection may be unreliable.



(a) Error Image (demosaiced - mosaiced)

(b) Binarized Error Image (demosaiced - mosaiced)

Figure 6: Error Images



(a) Localized Error Image with Bounding Box

(b) Bounding Box Transposed on Input Image

Figure 7: Bounding Box on Non-Native Content

Method 2: Detection and Localization with Periodic Properties of Interpolation and FCNN

As shown in Figure 2, overlapping patches of size 64×64 will be extracted from the input image. After that, a feature vector will be calculated for each patch. Then a fully connected neural network (FCNN) will be used to classify every pair of these patches and assign a group index to every patch based on the output of the classifier. Details for each major step are as follows.

Feature Extraction

We adopt the idea in [5] to calculate feature vectors for image patches. The main idea in this paper is that, after an interpolation process, there will be noticeable peaks in the extracted feature of patches from the image, when compared to the original version. As shown in Figure 8, the feature extraction process involves 4 major steps: The first step is selecting region of interest, by sliding a window over the image in raster order with step size N. The second step is called signal derivative computation. It is about computing the derivative image and edge detection. The third step is radon transformation. In this step, the projection will be calculated from 0 to 179 degrees in 1 degree increment. The final step is searching for periodicity in the projection by calculating its fast Fourier transform. In the paper, the gradient for projection at each angle is calculated before computing the FFT in order to emphasize the periodicity. We extract feature vectors from every image patch in a way similar to that introduced in [5] and made some minor changes in the first and second step to adapt the method to our settings. In the first step, we used a smaller patch size of 64×64 instead of 128×128 for higher accuracy of localization. As for the second step, we used a 3×3 Laplacian filter instead of an approximate derivative operator for edge detection.

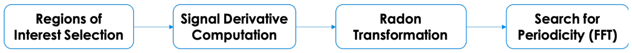

Regions of Interest Selection
Signal Derivative Computation
Radon Transformation
Search for Periodicity (FFT)

Figure 8: Steps of Patch Feature Extraction

Classifier Based on Fully Connected Neural Network

With the extracted features, we then use the FCN model shown in Figure 9 to classify feature pairs. The fully connected neural network in our proposed methods consists of 5 fully connected layers, with Sigmoid non-linear functions in between. The purpose for this FCN model is to classify whether the pair of input features are from patches that have undergone different types of geometric transformation. If they are different, the expected prediction will be 1 and if they are not, the expected prediction will be 0.

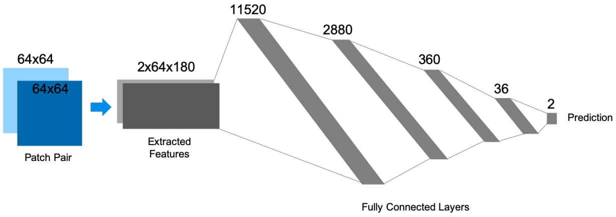


Figure 9: Fully Connected Neural Network Structure

As illustrated in Figure 9, the input to this FCN model is a pair of features of size 64 × 180 stacked together, which will be a 3-dimensional input of size $2 \times 64 \times 180$. Going through five fully connected layers with Sigmoid functions in between, the output will become a 1×2 vector representing a binary classification.

The procedures used to train the model are illustrated in Figure 10. To train the model, we used raw images from the UCID dataset [21] to build our own training dataset. To be more specific, we cut the raw images into 64 × 64 overlapping patches with step size 16. We then randomly choose one of the transformations to apply to each patch. The potential types of transformation are scaling, rotation, JPEG compression and skewing. To form a training sample, two different patches will be randomly selected from all the preprocessed patches we generate. Features are then calculated for the pair of patches and the corresponding labels will be created. Finally, during training, the model will be trained on 500 batches of training samples in every epoch and it will be trained until convergence.

In this paper, we set the batch size to be 50; and we use cross-entropy loss to regulate the model.

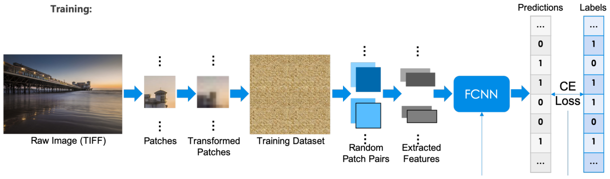


Figure 10: Training the FCNN

Detection and Localization

As the first step, an input image will be split into overlapping patches of size 64 × 64 with step size 16. After that, using the FCN model that we have trained, patches will be classified into different groups, based on their underlying transformations. To be specific, all patches will be assigned to group 0 initially. Then starting from group $k = 0$, the first patch in the group k will be paired with all following patches in the same group, and pairs of features extracted from patches in group k will be fed to the FCNN. If the prediction is 1, which means they belong to different types of transformation, then the latter patch in the pair will be classified to group $k+1$. We then repeat the process for group $k+1$ and so on. After that, the classes that only have one patch will be removed because they are mostly just noise due to the limitation of the FCNN. In the end, the algorithm determines whether the image contains non-native content by checking the number of remaining groups. If there are more than two groups, then the minority classes are very likely non-native contents. A bounding box will then be created to localize all patches in the classes that have the least number of patches.

Experimental Results

To test our algorithms, we collected a total of 96 images from Poshmark.com as our testing dataset. Within the 96 images, there are 46 images with non-native contents and 50 images without non-native contents. Some outputs are shown in Figures 11 and 12 in the APPENDIX section .

Confusion matrices for the detection accuracy of both methods are shown in Tables 1and 2. A summary of the overall accuracy, precision, and recall for both methods is shown in Table 3. Due to the limitation of our FCNN model and the lack of real training data, the FCNN model fails on some image patches in the testing samples that do not appear in the training data. As a result, the second method is too sensitive and has a comparably lower precision. The first method has the limitation of improper MSER region detection which also makes it insensitive to images that do not contain text.

	Predicted Positive	Predicted Negative	Total
Actual Positive	36	10	46
Actual Negative	15	35	50
Total	51	45	96

Table 1: Confusion Matrix for Method 1

	Predicted Positive	Predicted Negative	Total
Actual Positive	34	12	46
Actual Negative	30	20	50
Total	64	32	96

Table 2: Confusion Matrix for Method 2

	Accuracy	Precision	Recall
Method 1	73.96%	70.59%	78.26%
Method 2	56.25%	53.13%	73.91%

Table 3: Overall Accuracy, Precision and Recall for Both Methods

Conclusion

In this paper, we proposed two methods for detection and localization of non-native contents in online fashion images. Our experiments successfully demonstrated the effectiveness of both our proposed methods. The first method is adept at recognizing non-native text in a natural image but fails to do so when the entire image is non-native since a camera with a mosaicing algorithm was not used to capture that image. Furthermore, incorrect MSER region detection can also lead to natural images being categorized as images with non-native content. For the second method, due to the limitation of our FCNN model and the lack of real training data, the precision is limited.

Acknowledgments

We would like to thank all colleagues in our team who have provided assistance.

References

[1] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, Feb 2005.

[2] S. Prasad and K. R. Ramakrishnan, “On resampling detection and its application to detect image tampering,” in *2006 IEEE International Conference on Multimedia and Expo*, July 2006, pp. 1325–1328.

[3] S.-J. Ryu and H.-K. Lee, “Estimation of linear transformation by analyzing the periodicity of interpolation,” *Pattern Recognition Letters*, vol. 36, pp. 89 – 99, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551300370X

[4] A. C. Gallagher, “Detection of linear and cubic interpolation in jpeg compressed images,” in *The 2nd Canadian Conference on Computer and Robot Vision (CRV’05)*, May 2005, pp. 65–72.

[5] B. Mahdian and S. Saic, “Blind authentication using periodic properties of interpolation,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, Sep. 2008.

[6] B. Bayar and M. C. Stamm, “Design principles of convolutional neural networks for multimedia forensics,” in *Media Watermarking, Security, and Forensics*, 2017.

[7] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2016, pp. 1–6.

[8] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. S. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson, “Detection and localization of image forgeries using resampling features and deep learning,” *CoRR*, vol. abs/1707.00433, 2017. [Online]. Available: http://arxiv.org/abs/1707.00433

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions.” in *BMVC*, P. L. Rosin and A. D. Marshall, Eds. British Machine Vision Association, 2002. [Online]. Available: http://dblp.uni-trier.de/db/conf/bmvc/bmvc2002.html#MatasCUP02

[10] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, “Robust text detection in natural images with edge-enhanced maximally stable extremal regions,” in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 2609–2612.

[11] A. Gonzlez, L. M. Bergasa, J. J. Yebes, and S. Bronte, “Text location in complex images,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 617–620.

[12] L. Neumann and J. Matas, “Real-time scene text localization and recognition,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3538–3545.

[13] Y. Li and H. Lu, “Scene text detection via stroke width,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 681–684.

[14] A. Hooda, M. Kathuria, and V. Pankajakshan, “Application of forgery localization in overlay text detection,” in *ICVGIP*, 2014.

[15] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the sobel operator,” *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, April 1988.

[16] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[17] S. Guangling, S. Zhoubiao, and C. Yuejun, “Color filter array synthesis in digital image via dictionary re-demosaicing,” in *2010 International Conference on Multimedia Information Networking and Security*, Nov 2010, pp. 898–901.

[18] S. Cashmore, “An introduction to electronic imaging for photographers. adrian davies and phil fennessy oxford, uk: Focal press. isbn 0 240 51384 3. 1994. 125 pp. softback with photo cd. 19.99,” *Journal of Audiovisual Media in Medicine*, vol. 18, no. 4, 1995.

[19] O. Losson, L. Macaire, and Y. Yang, “Comparison of color demosaicing methods,” *Advances in Imaging and Electron Physics*, vol. 162, no. C, pp. 173–265, 2010.

[20] R. Malvar, L.-w. He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *International Conference of Acoustic, Speech and Signal Processing*. Institute of Electrical and Electronics Engineers, Inc., May 2004.

[21] G. Schaefer and M. Stich, “Ucid: An uncompressed color image database,” in *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307, 01 2004, pp. 472–480.

Author Biography

Karthick Shankar is an undergraduate senior at Purdue University studying Computer Engineering. His focus is primarily in the side of machine learning in image processing applications.

Litao Hu received his BS in Electronic Engineering from the Hong Kong University of Science and Technology (2017) and is current a PhD candidate in Electrical and Computer Engineering of Purdue University. As a research assistant in the Electronic Imaging System Laboratory, his research interests include image processing, machine learning, and deep learning.

APPENDIX

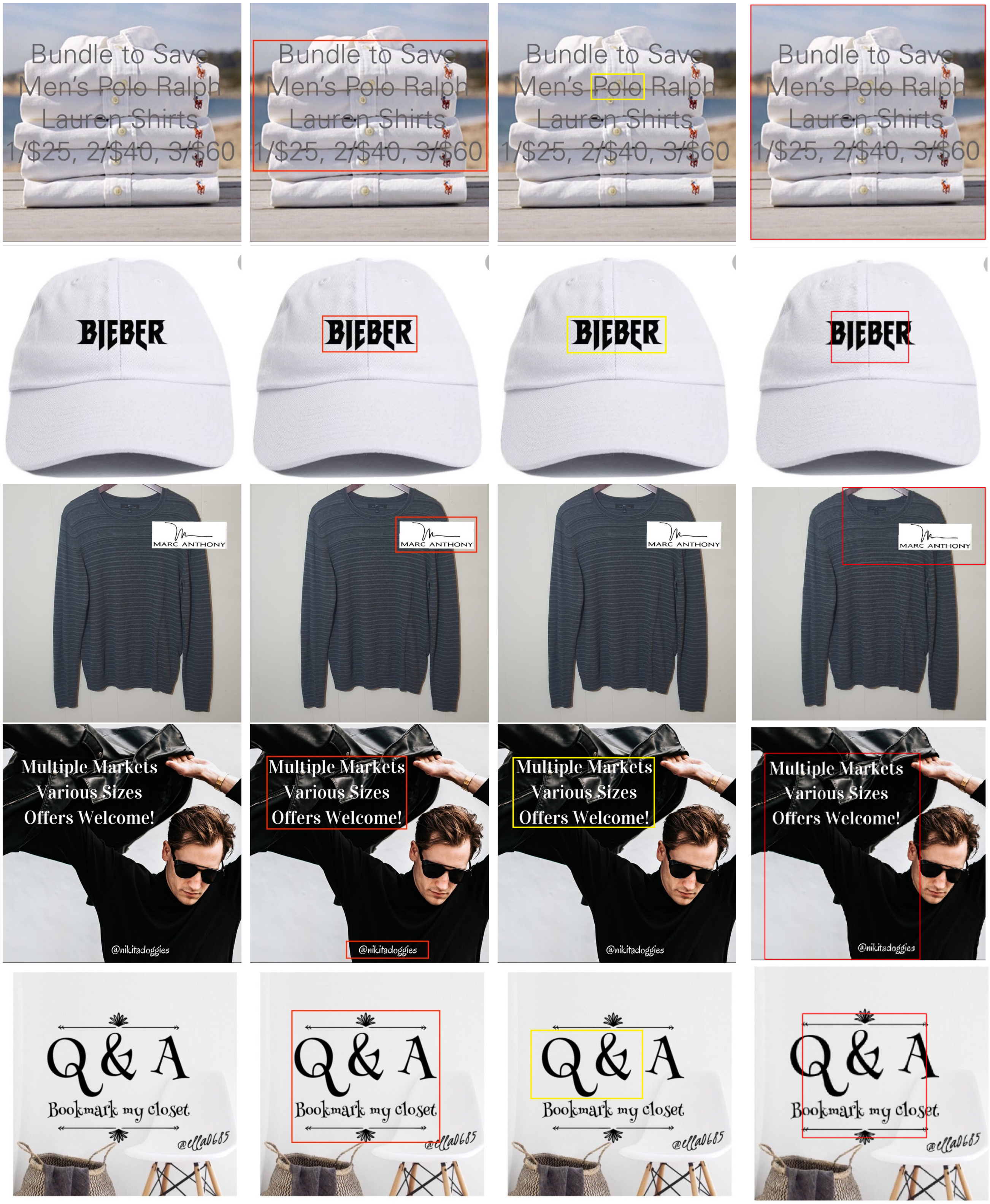


Figure 11: (First Column) Input Frame 1; (Second Column) Ground Truth; (Third Column) Method 1 Outputs; (Fourth Column) Method 2 Outputs

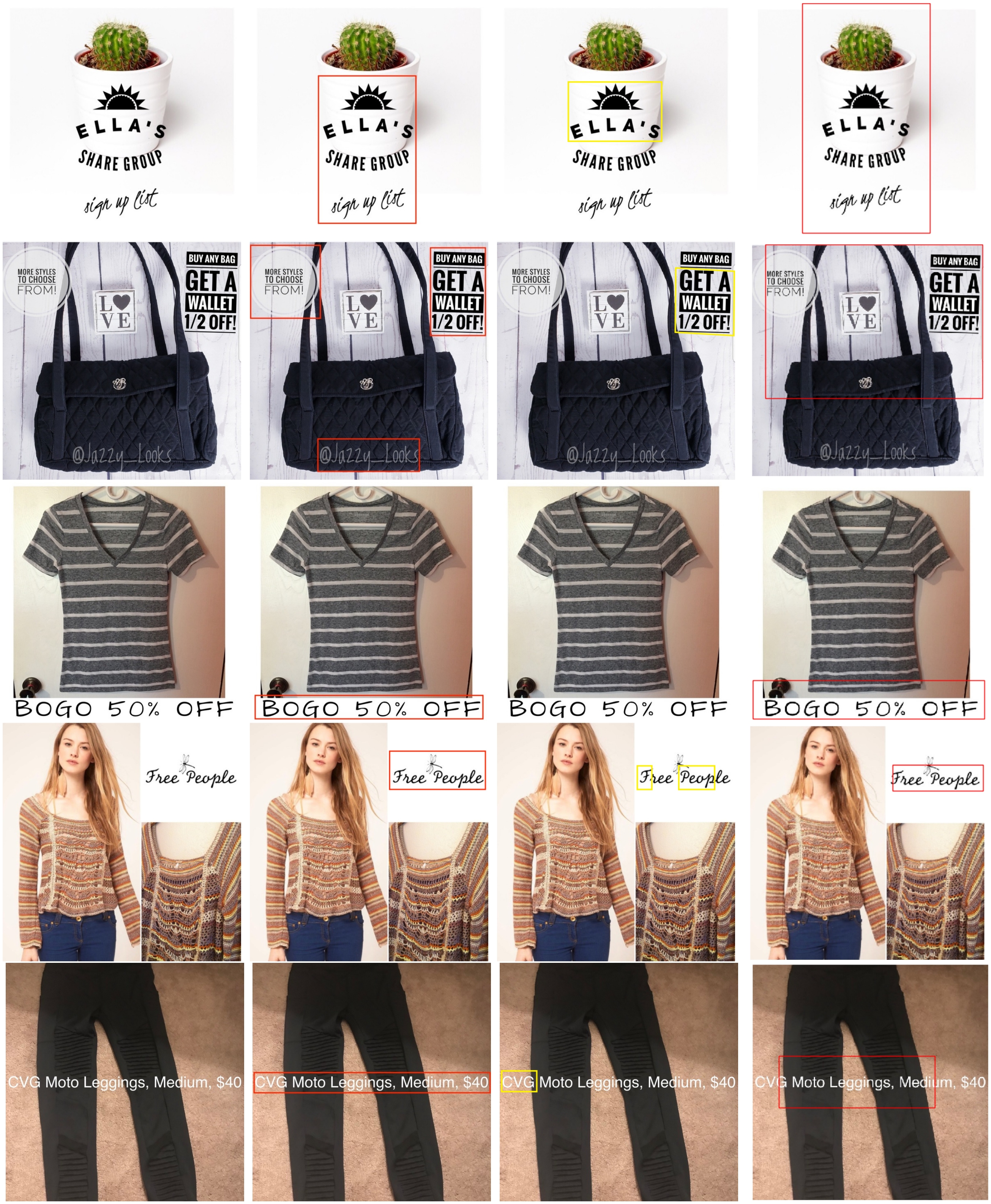


Figure 12: (First Column) Input Frame 1; (Second Column) Ground Truth; (Third Column) Method 1 Outputs; (Fourth Column) Method 2 Outputs

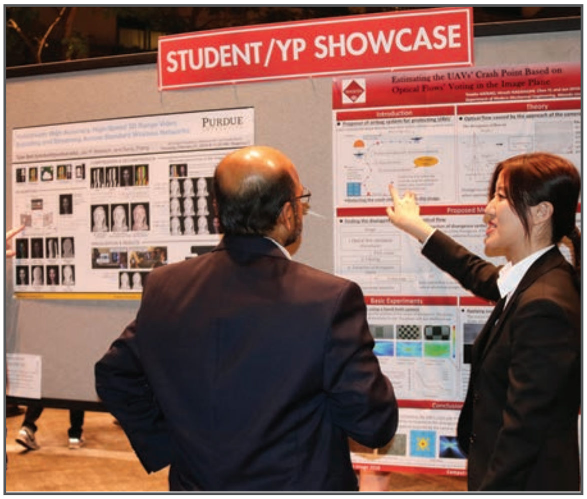
JOIN US AT THE NEXT EI!

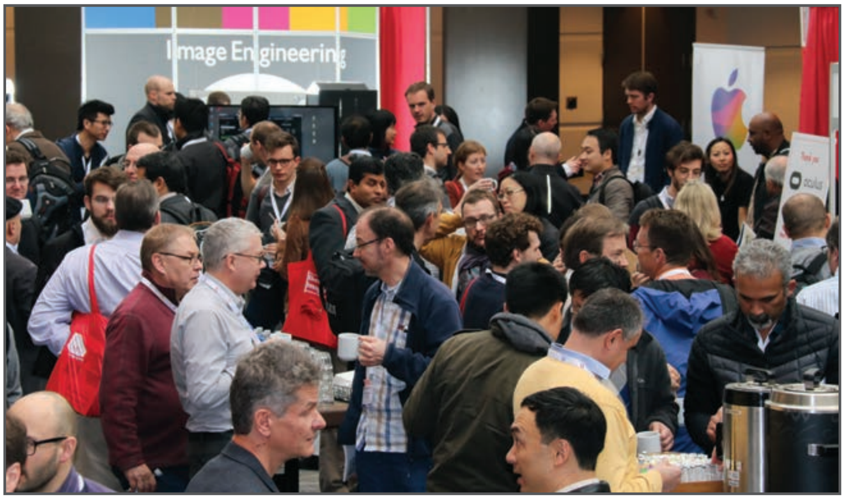
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





AR is a fundamental and transformative interface and platform.
20+ years from my initial working demo to today's AR displays
Optimistic that it won't take another 20 to reach ubiquitous consumer AR!
My first working AR prototype
Today's platforms
Ubiquitous consumer AR

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org